AFRL-AFOSR-VA-TR-2016-0253

YIP Formal Synthesis of Software-Based Control Protocols for Fractionated, Composable Autonomous Systems

Richard Murray
CALIFORNIA INSTITUTE OF TECHNOLOGY
1200 E. CALIFORNIA BLDV
PASADENA, CA 91125

07/08/2016
Final Report

Air Force Research Laboratory
AF Office Of Scientific Research (AFOSR)/RTA2

Arlington, Virginia 22203
Air Force Materiel Command

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. REPORT DATE *(DD-MM-YYYY)*	2. REPORT TYPE	3. DATES COVERED *(From - To)*
08-07-2016	FinalPerformance	01 Jun 2012 to 31 May 2015

4. TITLE AND SUBTITLE
YIP Formal Synthesis of Software-Based Control Protocols for Fractionated, Composable Autonomous Systems

5a. CONTRACT NUMBER

5b. GRANT NUMBER
FA9550-12-1-0302

5c. PROGRAM ELEMENT NUMBER
61102F

6. AUTHOR(S)
Richard Murray

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
CALIFORNIA INSTITUTE OF TECHNOLOGY
1200 E. CALIFORNIA BLDV
PASADENA, CA 91125 US

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
AF Office of Scientific Research
875 N. Randolph St. Room 3112
Arlington, VA 22203

10. SPONSOR/MONITOR'S ACRONYM(S)
AFRL/AFOSR RTA2

11. SPONSOR/MONITOR'S REPORT NUMBER(S)
AFRL-AFOSR-VA-TR-2016-0253

12. DISTRIBUTION/AVAILABILITY STATEMENT
A DISTRIBUTION UNLIMITED: PB Public Release

13. SUPPLEMENTARY NOTES

14. ABSTRACT
This project develops methods and tools for formally synthesizing distributed, softwarebased control protocols for autonomous systems. It tackles the challenge of establishing trust in autonomous systems through a shift from the traditional design+verify approach to specify+synthesize. Specifically, it focuses on fractionated system architectures, where heterogeneous modules delivering distinct services are composed into a functional system while sharing computing and power resources across networks. The architectural constraints due to fractionation are critical enablers of our strategy shift toward formal synthesis. A specify+synthesize design flow begins with formal specification of system requirements, architectural constraints, and information flow patterns. These specifications are automatically compiled into control protocols that utilize multiscale models of the system and measurements of its dynamic environment in order to realize these specifications.

15. SUBJECT TERMS
Stochastic Analysis, Information Systems

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			LAWTON, JAMES
Unclassified	Unclassified	Unclassified	UU		

Standard Form 298 (Rev. 8/98)
Prescribed by ANSI Std. Z39.18

				19b. TELEPHONE NUMBER *(Include area code)* 703-696-5999

YIP Formal Synthesis of Software-Based Control Protocols for Fractionated, Composable Autonomous Systems

California Institute of Technology
AFOSR Contract # FA9550-12-1-0302
Performance Period: June 1, 2012 – May 31, 2015

Prepared by:

Dr. Ufuk Topcu
The University of Texas at Austin
utopcu@utexas.edu
June 11, 2016

Abstract

This project develops methods and tools for formally synthesizing distributed, software-based control protocols for autonomous systems. It tackles the challenge of establishing trust in autonomous systems through a shift from the traditional "design+verify" approach to "specify+synthesize." Specifically, it focuses on fractionated system architectures, where heterogeneous modules delivering distinct services are composed into a functional system while sharing computing and power resources across networks. The architectural constraints due to fractionation are critical enablers of our strategy shift toward formal synthesis. A "specify+synthesize" design flow begins with formal specification of system requirements, architectural constraints, and information flow patterns. These specifications are automatically compiled into control protocols that utilize multiscale models of the system and measurements of its dynamic environment in order to realize these specifications.

The project has three thrusts: (i) Synthesis of embedded, reactive control protocols that account for heterogeneity in dynamics and requirements, measurement-based reconfiguration in dynamically changing adversarial as well as cooperative environments, faults and latency in communication and computing. (ii) Developing computable robustness characterizations and metrics for these protocols. (iii) Composing systems from reusable components, and incrementally constructing system-level performance and robustness certificates from a library of subsystem certificates.

Contents

1 Objectives

The objective of this project is to develop methods and tools for formally synthesizing distributed, software-based control protocols for autonomous systems. Establishing trust in

autonomy has increasingly become the bottleneck in the development and integration of such systems, and we tackle this challenge through a shift from the traditional "design+verify" approach to "specify+synthesize." Specifically, we seek answers to how we (i) can specify system requirements, architectural constraints, and information flow patterns, and (ii) synthesize control protocols that utilize multiscale models of the system and measurements of its dynamic environment in order to realize these specifications.

We focus on fractionated, composable system architectures, where heterogeneous, reusable modules delivering distinct services are composed into a functional system while sharing distributed computing and power resources across networks. The architectural constraints (on the physical coupling and information exchange between the modules) due to fractionation are critical enablers of our strategy shift—which would be impractical if naively applied to traditional monolithic systems with no separation between modules—toward formal synthesis. The project has three thrusts:

Thrust I – Synthesis of hierarchical, embedded control protocols: How can we synthesize reactive control protocols that account for (a) hybrid dynamics, heterogeneity in requirements, and mixed-criticality; (b) execution-time, measurement-based reconfiguration in dynamically changing adversarial as well as cooperative environments; (c) resource constraints; and (d) faults and latency in communication and computing?

Thrust II – Computable robustness metrics: What are suitable metrics that characterize robustness of the hierarchical control protocols against uncertainties in continuous/discrete models and due to sensing limitations, and violations in the underlying assumptions (e.g., on environment behavior)?

Thrust III – Composition of systems from reusable component libraries: How can we incrementally construct system-level certificates of performance and robustness from a library of subsystem certificates? How can we create such libraries?

The ability to synthesize—rather than merely analyze—the behaviors of subsystems (from Thrust I) and to assess their robustness (from Thrust II) increases the feasibility of compositional construction (in Thrust III). For successful execution, we merge and establish innovative connections between ideas from a broad background, including specification languages and formal methods in computer science, game theory, optimization-based, robust control, and graph and partial order theories.

The outcomes of this work include methods, computational tools, and insights needed to develop new, scalable design procedures that are aligned with the emerging characteristics of Air Force systems: highly autonomous with wide range of capabilities delivered in contested environments. The project offers a number of case studies as proofs of concept for the potential reductions in computational costs and improved applicability of formal specifications and synthesis for autonomous systems.

2 Accomplishments

We report the accomplishments in each of the three thrusts of the project.

2.1 Thrust I – Synthesis of hierarchical, embedded control protocols

Hierarchical and Compositional Synthesis with Parametric Reactive Controllers

Although automatic synthesis of realistic systems with large state spaces seems to be

currently unattainable, in practice, complex systems are often not constructed from scratch (an implicit assumption in many of the related works) but from a set of existing building blocks. For example, networked systems are built from a number of building block and, in robot motion planning, a robot usually has a number of predefined motion primitives that can be selected and composed to enforce a high-level objective. Intuitively, a compositional approach that solves smaller and more manageable subproblems, and hierarchically composes the solutions to implement more complicated behaviors seems to be a more plausible way to synthesize complex systems.

To this end, we developed a compositional and hierarchical framework for synthesis from a library of *parametric* and *reactive* controllers [C1]. Parameters allow us to take advantage of the symmetry in many synthesis problems, e.g., in motion planning for autonomous robots and vehicles. Reactivity of the controllers takes into account that the environment may be dynamic and potentially adversarial. We showed how these controllers could be synthesized from parametric objectives specified by the user to form a library of parametric and reactive controllers. We also provided a synthesis algorithm that selected and instantiates controllers from the library in order to satisfy a given safety and reachability objective. We implemented and applied the methods to an autonomous vehicle case study, where a controller was synthesized from a library of parametric and reactive controllers to safely navigate a controlled vehicle to its destination while avoiding collision with other uncontrolled vehicles.

Verification of Nonlinear Systems Using Automata Theory and Barrier Certificates

We developed a sound but incomplete method for the computational verification of specifications expressed in temporal logic against the behavior of dynamical systems evolving over (potentially partially) continuous state spaces [J1]. This new method merges ideas from automata-based model checking with those from control theory including so-called barrier certificates and optimization-based search for such certificates. More specifically, we considered linear temporal logic (excluding the "next" operator) formulas over atomic propositions that capture (sub)set memberships over the continuous state space. Under mild assumptions, the properties of the trajectories, which are salient for the verification, of the system can be characterized by infinite sequences (we call them traces) that track the atomic propositions satisfied along the corresponding trajectories (i.e., the subsets visited along the trajectory). Then, an automaton representation of the negation of the temporal logic formula guides a decomposition of the verification task into a finite collection of simpler constraints over the continuous state space. The satisfaction of these constraints in turn can be (potentially conservatively) proved by appropriately constructed barrier certificates.

The new method avoids explicit abstractions of the dynamics. On the other hand, the automaton representation of the specification may be interpreted as a "minimal" finite-state abstraction required for verification. The details due to the dynamics ignored in this abstraction are then accounted for by the barrier certificates only to the level of fidelity and locally over the regions of the continuous state space dictated by the dynamics. Similar to existing approaches for verifying nonlinear systems against temporal logic specifications, our approach is also not complete.

The method is in principle applicable to a broad family of dynamical systems as long as certain, relatively mild smoothness conditions hold. The step, which practically determines the applicability, of the proposed procedure is the computational search for barrier certificates. In this step, we focused on polynomial vector fields and utilized a combination of generalizations

of the S-procedure and sum-of-squares relaxations for global polynomial optimization. These techniques are relatively standard now in controls and have been used in other work on quantitative analysis of nonlinear and hybrid systems.

Automaton-Guided Controller Synthesis for Nonlinear Systems with Temporal Logic

Inspired by the automaton-guided verification results discussed above, we considered the problem of automatically synthesizing controllers for discrete-time nonlinear systems with temporal logic task specifications [C2]. Common approaches to temporal logic motion planning construct a finite discrete abstraction of the dynamical system. Instead of blindly doing expensive reachability computations to construct an abstraction of a dynamical system, the method we developed uses a coarse abstraction of the system and perform constrained reachability checks as needed for the task. It first creates an existential abstraction, i.e., a finite abstraction of the system where transitions between abstract states are assumed to exist, but have yet not been verified (e.g., through computationally expensive set-to-point reachability computations) to exist in the system. It then creates a product automaton from the finite-state abstraction and an automaton representing the underlying specification. The product automaton guides reasoning about complicated temporal logic properties as a sequence of simple temporal properties that can be analyzed using constrained reachability techniques. This sequence of constrained reachability problems is called an abstract plan. However, the system might not be able to follow a given abstract plan since dynamic constraints were not considered in the existential abstraction. The next step is checking the abstract plan with the continuous dynamics by solving a sequence of constrained reachability problems. If this sequence is infeasible, the product automaton is updated and a new abstract plan is generated.

The resulting method is independent of the specific techniques used to compute constrained reachability, and is amenable to parallel computing. Despite the frameworkÕs generality, it is also computationally powerful, as we showed with examples that improve on state-of-the-art techniques for temporal logic motion planning for high-dimensional (10+ states) continuous systems.

2.2 Thrust II – Computable robustness metrics

Synthesizing Robust Discrete Controllers under Modeling Uncertainty

Robustness—a system's ability to function correctly under uncertainties, for example, due to imperfections in the way the evolution of the system and its interactions with its environment are modeled—is a key attribute to predictable operation (and graceful failure). Though well-studied for physical engineering artifacts, it has been hardly explored for distributed embedded systems. Approaching this from a computer science perspective, a reason for the lack of suitable robustness notions is that computing systems are conveniently modeled as discrete mathematical objects with no underlying (non-trivial) topology where uncertainties and their impact can be quantified. Furthermore, even though controls have explicitly modeled such uncertainties and developed dedicated methods and tools, they have been limited to rather restrictive representations and cannot directly address the critical interplay between physical components and computing/communication.

As a step toward addressing the need for characterizations and computable metrics to support the analysis, design, and construction of robust embedded control systems, we investigted robustness of discrete, reactive control protocols synthesized to guarantee system's correctness with respect to given temporal logic specifications [C3]. We considered uncertainties in open

finite transition systems, i.e., transition systems with uncontrolled inputs, due to unmodeled transitions. We reformulated the resulting robust synthesis problem as a temporal logic game. In particular, we utilized specifications that belong to the so-called generalized reactivity [1] (GR[1]) fragment of linear temporal logic for which there exist polynomial complexity solvers. We showed that if the specification is a GR[1] formula, so is the augmented specification in the resulting robust synthesis problem. Hence, robustification of protocol synthesis with the specific uncertainty model does not change its complexity class.

Resilience to Intermittent Assumption Violations in Reactive Synthesis

Automatically synthesizing reactive systems from their specifications is an attractive alternative to constructing these by hand. Even when a complete specification is not available, formal synthesis is a useful approach to analyze the specifications for the parts of the system to be constructed and allows to explore the design alternatives in a structured way. To fully benefit from synthesis technology, measures have to be taken to ensure that the implementations computed in the process are of good quality. Example quality criteria include energy consumption, size of the implementation, and the resilience of the implementation against changes in the conditions under which the system operates. Intuitively, the latter criterion means that a system should work as well as possible in scenarios in which the assumptions about its environment are violated. In other words, the system shall degrade gracefully, and all safety-relevant properties of the system should be fulfilled "whenever" possible.

While it would obviously be best if a synthesized system does not rely on any environment assumptions being satisfied, this is typically not possible. For example, if we require a robot to go from one point in a workspace to another point and there is an obstacle in between, then the assumption that the position of the robot is updated according to its actions (move left, move right, etc.) needs to be made. Manually constructed reactive system controllers typically only rely on these environment assumptions being satisfied in situations in which they are crucially needed. A reactive synthesis procedure on the other hand will typically compute a controller that lets it come very close to the obstacle, and thus the resulting controller cannot even tolerate a single "glitch" in the environment assumptions. The reason for this is by default, reactive synthesis procedures do not optimize towards controller behavior that allows for error-resilience (such as staying away from an obstacle as far as possible). Even worse, once an assumption is violated, the system is free to behave in an arbitrary manner and in particular, possibly fail to fulfill its requirements. The violation of assumptions in the field does also not necessarily mean that they have been modeled incorrectly, as typically not all eventualities can be modeled correctly and precisely, like components of the robot breaking at runtime or dirt on sensors leading to imperfect measurements.

We formulated and investigated the problem of synthesizing error-resilient systems from specifications in temporal logic [C4]. We concentrated on the generalized reactivity(1) fragment of linear-time temporal logic, for which an efficient and symbolic synthesis algorithm is known. We showed how to add the requirement of being k-resilient to such a specification. That is the system has to tolerate arbitrarily many violations of safety assumptions ("glitches"), as long as in between every k such glitches, there is a long enough period in which no glitch occurs so that the system can recover from the earlier k glitches. By not exceeding the class of generalized reactivity(1) specification in this process, the resulting method ensures that also the synthesis problem for the resulting specification can be solved efficiently.

Automatically synthesizing error-resilient systems enables to effectively perform design space

exploration: compute (i) which assumptions need to be seen as strict, i.e., need to be satisfied all of the time, (ii) for which assumptions arbitrarily many glitches can be tolerated, and (iii) for which assumption some glitches can be tolerated, and whose violations should count towards the value of k. We developed an exploration algorithm that searches for all Pareto-optimal assignments of the assumptions to these categories that represent implementable error-resilience guarantees. Searching for these optimal solutions gives the system designer the insight of what the specifications imply with respect to the systemÕs resilience and thus to select the most reasonable solution for the practical application in mind, without the need to formalize the preferences in advance. Additionally, it allows the system designer to state the assumptions in a very conservative manner whenever a precise model of the assumptions cannot be given.

2.3 Thrust III – Composition of systems from reusable component libraries

High complexity of synthesis procedures has restricted their application to relatively small- to modest-sized problems. The pioneering work by Pnueli showed that reactive synthesis from linear temporal logic specifications is intractable which prohibited the practitioners from utilizing automated synthesis algorithms in practice. Recent advances in this growing research area have enabled automatic synthesis of interesting real-world systems indicating the potential of the synthesis algorithms for solving realistic problems. The key enabling factors we pursue in this thrust are (i) focusing on subclasses and practically relevant examples that are practically relevant yet offer improved computational characteristics and (ii) exploiting the underlying modular structures in the system.

Distributed Power Allocation for Vehicle Management Systems

Vehicle management systems (VMS) control and coordinate a number of subsystems of aerial vehicles, e.g., flight controllers, electrical systems, fuel management, environmental control systems, deicing units, and landing gear, and interface with additional aircraft subsystems, e.g., sensor pointing, data acquisition, and pilot and ground interfaces. Traditional VMS are typically based on federated architectures in which integrated hardware and software components realize independent or loosely interconnected functions. Next-generation VMS are expected to incorporate distributed computation, advanced networking, and increased levels of autonomous operations to manage physical resources, e.g., requirements on electric power generation and flows. Additionally, the move to autonomous flight will require the VMS to be interactive in dynamically changing environments and reconfigurable. Integrated modular avionics (IMA) architectures driven by these trends provide an alternative to federated architectures. The IMA architectures utilize high integrity, partitioned platforms that host multiple avionics functionalities of different criticalities. The IMA architecture is based on highly-integrated resource management among the functionalities that share the existing resources and leads to two competing trends: possibilities for system-level optimization by dynamically allocating resources (potentially leading to reductions in weight) come at the expense of extra layers of integration complexities.

Due to the increasing complexity of VMS, certification of safety and performance properties will necessitate use of formal specifications. Furthermore, systematic methods for verifying systems against these specifications and alternatively for synthesizing correct-by-construction control protocols will likely reduce the amount of costly validation experiments and tests.

Motivated by these challenges, we investigated distributed synthesis of distributed control protocols that enable dynamic configuration for integrated power management in VMS [J2, C5]. In particular, we focused on electric power management on “more-electric” aircraft between

a subset of the subsystems, namely flight controllers, deicing units, and environmental control. The main design considerations in our formulation included the following: (i) Real-time reconfiguration in a dynamic environment: The subsystems interact with their environment (e.g., due to outside temperature variations and changes in flight conditions); hence, they need to react to the changes in their environment in real time. (ii) Fault tolerance: The power management systems should be able to reconfigure in the presence of faults or failures to satisfy its safety and performance requirements. (iii) Resource constraints: With the increase in the electric loads and introduction of integrated architectures, the subsystems share limited electric power resources. (iv) Mixed-criticality subsystems: The subsystems have varying levels of flight-criticality, e.g., flight controllers are highly critical whereas environmental control is of lower criticality. Therefore, the control protocol for power management needs to account for the prioritization of the loads from these subsystems while maintaining non- flight-critical criteria, e.g., certain measures of passenger comfort, within acceptable bounds.

We utilized linear temporal logic as a formal specification language and developed techniques for compositional design of correct-by-construction, distributed protocols for dynamic configuration of integrated power management. Distribution of both the design and the implementation is considered to facilitate modularity of design, e.g., in contract based design, to reduce onboard power and information flows, and to alleviate the computational complexity of the synthesis procedure. The output of the synthesis procedure is a hierarchical control protocol with a discrete planner responsible for the satisfaction of certain high-level specifications and a continuous control that implements the discrete plan at the lower level. In this project, we focused on the high-level design problem; and assume that abstract discrete models of underlying continuous variables and dynamics are available. Such discrete models can be obtained by using abstraction techniques that have been proposed for representing the behavior of a continuous system with finitely many states and transitions among them which capture the relevant dynamics.

Our distributed design procedure is based on decomposing the global specification into local ones in order to enable distributed synthesis and implementation of local control protocols. The feasibility of the proposed distributed design procedure depends on the choice of the decomposition structure of the underlying system, the strength of the coupling (through the exchange of physical resources and information) between them, and the expressiveness of the interface models. In this work, we investigated three compositional synthesis settings, which, essentially, illustrate how refining the interface models, either by tightening the constraints on the inter-system flow of physical resources or by increasing the amount of information exchange, suppress the conservatism of distributed synthesis. The sub-project we discuss next automates such refinements of specifications.

Pattern-Based Assume-Guarantee Synthesis

Compositional synthesis techniques can potentially address the scalability problem by solving the synthesis problem for smaller components and merging the results such that the composition satisfies the specification. The challenge is then to find proper decompositions and assumptions and guarantees such that each component is realizable, its expectations of its environment can be discharged on the environment and other components, and circular reasoning is avoided, so that the local controllers can be implemented simultaneously and their composition satisfies the original specification.

In *pattern-based assume-guarantee synthesis* framework, we considered a system with two controllable agents reacting to their dynamically changing and adversarial environment

[C6, C7]. We decomposed the global specification into two local specifications, one for each agent. We refined the local specifications by automatic synthesis of assumptions and guarantees through analysis of strategies and counter-strategies obtained for the agents' local specifications. We showed how behaviors of the environment and the system could be inferred from counter-strategies and strategies, respectively, as formulas in special forms called *patterns*.

In this method, local specifications are refined until both become realizable, and under certain conditions, the strategies synthesized for the local specifications guarantee the satisfaction of the global specification. Intuitively, additional assumptions and guarantees synthesized during the refinement process are "contracts" between the agents that allow each of them to compute a strategy for its local specification while ensuring the satisfaction of the global specification for the system.

Compositional Synthesis of Reactive Controllers for Decoupled Multi-Agent Systems

In assume-guarantee synthesis , systems with multiple components can be treated in a *decentralized* manner by considering one component as a part of the environment of another component. However, in this reactive approach it is difficult to capture and model the need for joint decision-making and cooperative objectives. In order to address this difficulty, we developed a compositional framework for a special class of multi-agent systems (inspired by decentralized control and swarm robotics literature) based on automatic decomposition of objectives and compositional reactive synthesis using maximally permissive strategies [C8]. In this approach, we assumed that the objective of the system were given in conjunctive form. We made the observation that in many cases, each conjunct of the global objective only refers to a small subset of agents in the system. We took advantage of this structure to decompose the synthesis problem: for each conjunct of the global objective, we only considered the agents that were involved, and computed maximally permissive strategies for those agents with respect to the considered conjunct. We then intersected the strategies to remove potential conflicts between them, projected back the constraints to subproblems, solved them again with updated constraints, and repeated this process until the strategies reach a fixed point. With this approach we managed to solve synthesis problems for systems with multiple agents and objectives such as collision avoidance, formation control and reachability, and for grid-world of sizes that were much larger than the cases considered in similar works in the related literature. We showed that the compositional algorithm outperforms the centralized synthesis approach, both from time and memory perspective, and were able to solve problems for which the centralized algorithm was infeasible.

Optimal control in Markov decision processes via distributed optimization

Given a stochastic system modeled as a Markov decision process (MDP), the synthesis problem is to find a policy that achieves optimal performance under a given quantitative criterion regarding given temporal logic formulas. For instance, the objective may be to find a policy that maximizes the probability of satisfying a given temporal logic formula. In such a problem, we need to keep track of the evolution of state variables that capture system dynamics as well as predicate variables that encode properties associated with the temporal logic constraints. As the number of states grows exponentially in the number of variables, we often encounter large MDPs, for which the synthesis problems are impractical to solve with centralized methods. The insight for control synthesis of large-scale systems is to exploit the modular structure in a system so that we can solve the original problem by solving a set of small sub-problems.

We developed a distributed optimization method for large MDPs subject to temporal

logic constraints [C9]. We first introduced a decomposition method for large MDPs, and proved a property that the resulting decomposition supports the application of the proposed distributed optimization. For a sub-class of MDPs whose graph structures are planar graphs, we introduced an efficient decomposition algorithm that exploits the modular structure for the underlying MDP caused by loose coupling between subsets of states and its constituting components. Then, given a decomposition of the original system, we employed a distributed optimization method called block splitting algorithm to solve the planning problem with respect to discounted-reward objectives in large MDPs and average-reward objectives in large ergodic MDPs. Our method concurrently solves the set of sub-problems and penalizes the mismatches between their solutions in one step during each iteration. Since the distributed control synthesis is independent from the way how a large MDP is decomposed, any decomposition method can be used. We were able to apply the method on motion planning problems over grid worlds whose sizes are significantly larger (e.g., 1000-by-100) than those in relevant literature.

3 Personnel Supported

The project partly supported PI Topcu during his postdoctoral studies at California Institute of Technology and during this research faculty appointment at the University of Pennsylvania.

At the University of Pennsylvania, it supported a graduate student, Salar Moarref, who received his Ph.D. in 2016. Additionally, it provided partial support to a postdoctoral scholar, Shuo Han.

4 Technical Publications

4.1 Journal Publications

[J1] T. Wongpiromsarn, U. Topcu and A. Lamperski, "Automata theory meets barrier certificates: Temporal logic verification of nonlinear systems," IEEE Transactions on Automatic Control, 2015.

[J2] R. Rogersten, H. Xu, N. Ozay, U. Topcu, and R. M. Murray, "Synthesis and Validation of Control Software For A Vehicular Electric Power Distribution Testbed," Journal of Aerospace Information Systems, pp. 665–678, 2014.

[J3] J. Liu, N. Ozay, U. Topcu and R. M. Murray, "Synthesis of Reactive Switching Protocols From Temporal Logic Specifications," IEEE Transactions on Automatic Control, vol. 58, pp. 1771–1785, 2013.

4.2 Conference Publications

[C1] R. Alur, S. Moarref and U. Topcu. "Compositional Synthesis with Parametric Reactive Controllers," in the Proceedings of the Conference on Hybrid Systems: Computation and Control, 2016.

[C2] E. Wolff, U. Topcu and R. M. Murray, "Automaton-guided controller synthesis for nonlinear systems with temporal logic, in the Proceedings of the International Conference on Intelligent Robots and Systems, 2013.

[C3] U. Topcu, N. Ozay, J. Liu and R. M. Murray, "On Synthesizing Robust Discrete Controllers under Modeling Uncertainty," in the Proceedings of the Conference on Hybrid Systems: Computation and Control, 2012.

[C4] R. Ehlers and U. Topcu, "Resilience to Intermittent Assumption Violations in Reactive Synthesis," in the Proceedings of the Conference on Hybrid Systems: Computation and Control, 2014.

[C5] N. Ozay, U. Topcu and R. M. Murray, "Distributed power allocation for vehicle management systems," in the Proceedings of the Conference on Decision and Control, 2011.

[C6] R. Alur, S. Moarref and U. Topcu, "Pattern-Based Refinement of Assume-Guarantee Specifications in Reactive Synthesis," in the Proceedings of the Conference on Tools and Algorithms for the Construction and Analysis of Systems, 2015.

[C7] R. Alur, S. Moarref and U. Topcu, "Counter-strategy guided refinement of GR[1] temporal logic specifications," in the Proceedings of the Conference on Formal Methods in Computer-Aided Design, 2013.

[C8] R. Alur, S. Moarref and U. Topcu, "Compositional Synthesis of Reactive Controllers for Multi-Agent Systems," in the Proceedings of the Conference on Computer-Aided Verification, 2016.

[C9] J. Fu, S. Han and U. Topcu, "Optimal control in Markov decision processes via distributed optimization," in the Proceedings of the Conference on Decision and Control, 2015.

5 Interactions/Transitions

The project team organized technical workshops (taught broadly) and established close collaboration with AFRL researchers.

5.1 Workshops

- Specification, Design and Verification of Distributed Embedded Systems. A week-long short course taught at the European Embedded Control Institute, 2012 and 2013. (With Richard Murray and Tichackorn Wongpiromsarn.)
- Specification, Design and Verification of Distributed Embedded Systems. A two-day, hands-on workshop presented at Air Force Research Lab, 2013. (With Richard Murray.)
- Verification and Synthesis for Hybrid Systems. A short course taught at Air Force Research Lab, 2013. (With George Pappas.)

5.2 AFRL Visits

The PI established collaborations with AFRL researchers through a series of short-term visits to Dayton, OH over the period of 2013-2015.

6 Patent Disclosures

None

7 Honors

The PI was an Air Force Research Lab Summer Faculty Fellow in 2012.